Perspectives on Balanced Sequences

Jos H. Weber, Kees A. Schouhamer Immink, Paul H. Siegel, and Theo G. Swart

***Abstract*—We examine and compare several different classes of “balanced” block codes over q-ary alphabets, namely *symbol-balanced* (SB) codes, *charge-balanced* (CB) codes, and *polarity-balanced* (PB) codes. Known results on the maximum size and asymptotic minimal redundancy of SB and CB codes are reviewed. We then determine the maximum size and asymptotic minimal redundancy of PB codes and of codes which are both CB and PB. We also propose efficient Knuth-like encoders and decoders for all these types of balanced codes.**

***Index Terms*—coding theory, balanced codes, modulation codes, asymptotic redundancy**

I. Introduction

There are several different classes of block codes over a q-ary integer alphabet that can be described as being “balanced” in some sense. Consider, for example, the symmetric alphabets $\mathcal{A}_q = \{-q+1, -q+3, -q+5, \ldots, q-3, q-1\}$ that arise in the context of pulse amplitude modulation (PAM), e.g., $\mathcal{A}_4 = \{-3, -1, +1, +3\}$, $\mathcal{A}_5 = \{-4, -2, 0, +2, +4\}$. We say that a code is *symbol-balanced* (SB) over $\mathcal{A}_q$ if, in each codeword, all q alphabet symbols appear equally often. A *charge-balanced* (CB) code is one in which the sum of the symbols in each codeword is zero. We also define *polarity-balanced* (PB) codes, for which, in every codeword, the number of positive symbols equals the number of negative symbols. For q odd, this definition does not constrain the number of zero symbols.

It is easy to see that for $q = 2$, i.e., for bipolar sequences of even length n, these three notions of being “balanced” are completely equivalent. For $q = 3$, i.e., for sequences over the alphabet $\{-2, 0, +2\}$, the notions of CB and PB are equivalent, but the SB sequences form a proper subset of the set of CB and PB sequences. For example, the sequence $(-2, -2, +2, 0, -2, +2, +2, +2, -2)$ of length 9 is CB and PB, but not SB. For $q \geqslant 4$, all three notions are mutually distinct. Any sequence which is SB is also CB and PB, but there do exist sequences which are PB but not CB (e.g., $(-3, -1, +1, +1)$ over $\mathcal{A}_4$) and sequences which are CB but not PB (e.g., $(+3, -1, -1, -1)$ over $\mathcal{A}_4$). Furthermore, there exist sequences which are both CB and PB (denoted as CPB) but not SB (e.g., $(-3, -3, +3, +3)$ over $\mathcal{A}_4$). In conclusion, the general relationship among the balancing criteria discussed above can be represented by the Venn diagram shown in Fig. 1.

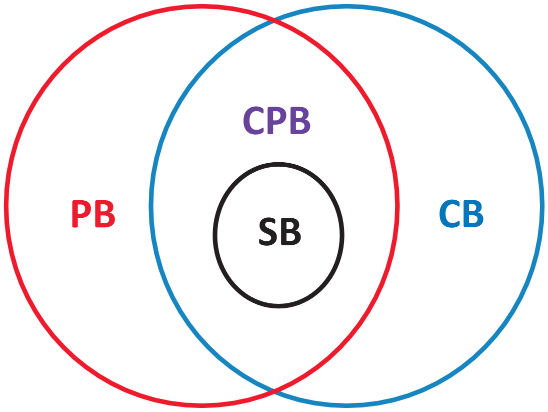

Fig. 1. Relationships among the symbol-balanced (SB), charge-balanced (CB), polarity-balanced (PB), and charge & polarity-balanced (CPB) properties.

Balanced codes have found applications in digital communications and data storage technology [7]. They have been widely studied in the literature, particularly for the binary case, e.g., [1], [3], [4], [9], [17], [18]. Some constructions also take into account error correction capabilities, e.g., [2], [13], [20], [22]. Results for non-binary alphabets have been presented for the SB and CB cases, albeit under different (or no specific) names, e.g., [11] (SB) and [6], [19] (CB). To the best of our knowledge, the PB concept for non-binary sequences is new and has not been studied before. It is of particular interest for applications which demand a balancing of positive and negative symbols, possibly in combination with a charge constraint. In this paper, we determine the number of q-ary PB sequences of length n as well as the number of q-ary sequences of length n which are CPB, i.e., both CB and PB. From this, we derive expressions for the minimum redundancy of PB and CPB codes, which are compared to the corresponding expressions for SB and CB codes.

A celebrated method to generate and decode bipolar balanced sequences of even length n was presented by Knuth [9]. The key idea is to invert the first z symbols of the information sequence such that the resulting sequence is balanced. Knuth showed that it is always possible to find at least one such balancing index z. By communicating the value of z through a (balanced) prefix, decoding can be performed by inverting the first z symbols of the coded sequence. The redundancy of this elegant method is roughly $\log_2(n)$, which is about twice the minimum and can thus be considered as a price to be paid for simplicity. In this paper, we extend Knuth’s

Submission to IEEE Transactions on Information Theory, September 28, 2012.

J. H. Weber is with Delft University of Technology, Delft, The Netherlands (e-mail: j.h.weber@tudelft.nl). K.A. Schouhamer Immink is with Turing Machines Inc., Rotterdam, The Netherlands (e-mail: immink@turing-machines.com). P.H. Siegel is with the Department of Electrical and Computer Engineering and the Center for Magnetic Recording Research, University of California, San Diego, La Jolla, CA 92093, U.S.A. (e-mail: psiegel@ucsd.edu). T.G. Swart is with the Department of Electrical and Electronic Engineering Science, University of Johannesburg, Auckland Park, South Africa (e-mail: tgswart@uj.ac.za).

P.H. Siegel was supported in part by the Center for Magnetic Recording Research at the University of California, San Diego. Portions of this work were done while J.H. Weber and P.H. Siegel were visiting the Centre Interfacultaire Bernoulli (CIB) at the Ecole Polytechnique Fédérale de Lausanne (EPFL) in Lausanne, Switzerland, in the autumn of 2011. Part of this work was presented at IEEE International Symposium on Information Theory, Seoul, Korea, 2009.

method, which assumes bipolar sequences, to larger alphabets. In particular, we present Knuth-like design methods for all balancing perspectives under consideration, i.e., for SB, CB, PB, and CPB.

The rest of this paper is organized as follows. In Section II, some definitions and preliminaries are presented. Then, in Section III, we first review known expressions for the maximum sizes of q-ary SB and CB codes of length n, as well as the minimal redundancy of these codes. We then derive the corresponding expressions for PB and CPB codes. In Section IV, we describe Knuth-like constructions for a variety of codes with various combinations of SB, CB, and PB properties. Finally, the paper is concluded in Section V.

II. Preliminaries

A. Alphabets and Balancing

In Section I, we introduced the alphabet

$$\mathcal{A}_q = \{-q+1, -q+3, -q+5, \ldots, q-3, q-1\},$$

where $q \geqslant 2$. We now formally define when a sequence $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in (\mathcal{A}_q)^n$ is balanced, for each of the considered perspectives.

- A sequence $\mathbf{x}$ of length $n = qm$, with $m \geqslant 1$, is *symbol-balanced* (SB) if all q symbols in $\mathcal{A}_q$ appear equally often in $\mathbf{x}$, i.e.,
$$|\{i : x_i = j\}| = m$$
for all $j \in \mathcal{A}_q$.
- A sequence $\mathbf{x}$ of length n, with n being a positive integer which is even if q is even, is *charge balanced* (CB) if the sum of all symbols in $\mathbf{x}$ is equal to 0, i.e.,
$$\sum_{i=1}^{n} x_i = 0.$$
- A sequence $\mathbf{x}$ of length n, with n being a positive integer which is even if q is even, is *polarity balanced* (PB) if the number of positive symbols in $\mathbf{x}$ equals the number of negative symbols, i.e.,
$$|\{i : x_i > 0\}| = |\{i : x_i < 0\}|.$$
- A sequence $\mathbf{x}$ of length n, with n being a positive integer which is even if q is even, is *charge and polarity balanced* (CPB) if it is both CB and PB.

Note that for lengths n which do not comply with the specifications, there exist no sequences satisfying the desired property. Hence, throughout this paper, we will assume that n is a multiple of q for SB codes and that, in case q is even, n is even for CB, PB, and CPB codes.

When studying q-ary balanced codes, other alphabets than $\mathcal{A}_q$ have also been considered in the literature, a prominent example being

$$\mathbb{Z}_q = \{0, 1, \ldots, q-1\}.$$

Also balanced codes over the roots of unity alphabet

$$\Phi_q = \{e^{2\pi i h/q} : h = 0, 1, \ldots, q-1\},$$

where $i = \sqrt{-1}$, have received quite some attention, e.g., [5], [12]. The choice of the alphabet may influence the balancing notion. This is not the case for symbol balancing, which is clearly independent of symbol representation. The number of SB sequences of a certain length n will be the same for any q-ary alphabet. The same conclusion is valid for polarity balancing, as long as we divide the alphabet symbols into two classes of equal size, with one neutral symbol in case q is odd. However, the notion of charge balancing is coupled to the choice of the alphabet. First of all, it demands that an additive operation is defined on the alphabet symbols, which, by the way, does not have to be closed with respect to the alphabet, i.e., a sum of alphabet symbols may take values outside the alphabet. The naming *'charge'* and the choice to fix the sequence symbol sum $\sum_{i=1}^{n} x_i$ to zero, as in the CB definition above, have been inspired by practical PAM-like applications. However, in other cases it may be desirable to fix the sum to another value. Also, the maximum number of CB sequences of a certain length may depend on the choice of the alphabet: for an irregularly spaced alphabet other results could be obtained than for a regularly spaced alphabet like $\mathcal{A}_q$.

Throughout this paper, we will assume that the code alphabet is $\mathcal{A}_q$. Still, many derived results on maximum code sizes, minimum redundancies, etc., are also valid for other alphabets. Particularly, when the alphabet can be obtained by applying a bijective mapping of the format

$$i \rightarrow ai + b \tag{1}$$

on the symbols from $\mathcal{A}_q$, where $a \neq 0$ and b are real numbers, then all results obtained for $\mathcal{A}_q$ also hold for the other alphabet (and vice versa), even the CB results. Note that $\mathbb{Z}_q$ is within this category (by choosing $a = -1/2$ and $b = (q-1)/2$). This implies that in $\mathbb{Z}_q$, the symbols smaller than $(q-1)/2$ should be called *'positive'* and the symbols larger than $(q-1)/2$ *'negative'*. Furthermore, the charge constraint should be replaced by $\sum_{i=1}^{n} x_i = n(q-1)/2$ in case the alphabet is $\mathbb{Z}_q$.

B. Codes and Redundancy

A *code* of length n is a set of sequences of length n. A code is said to be SB, CB, PB, or CPB if all codewords satisfy the respective properties. The sets of all SB, CB, PB, and CPB sequences of length n over $\mathcal{A}_q$ are denoted by $C_{\text{SB}}(n,q)$, $C_{\text{CB}}(n,q)$, $C_{\text{PB}}(n,q)$, and $C_{\text{CPB}}(n,q)$, respectively, and their sizes by $M_{\text{SB}}(n,q)$, $M_{\text{CB}}(n,q)$, $M_{\text{PB}}(n,q)$, and $M_{\text{CPB}}(n,q)$, respectively. The *redundancy* r of a q-ary code of length n and size M is

$$r = n - \log_q M. \tag{2}$$

The minimum redundancies of SB, CB, PB, and CPB codes of length n over $\mathcal{A}_q$ are denoted by $r_{\text{SB}}(n,q)$, $r_{\text{CB}}(n,q)$, $r_{\text{PB}}(n,q)$, and $r_{\text{CPB}}(n,q)$, respectively.

C. Stirling Approximation

In this paper, we will derive (asymptotic) expressions for the minimum redundancy. In the analysis we make frequent and implicit use of Stirling's approximation for factorials, stated here for convenience. For $n \geqslant 1$, it holds that

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\lambda_n}$$

where $\frac{1}{12n+1} \leqslant \lambda_n \leqslant \frac{1}{12n}$. Hence,

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + O\left(\frac{1}{n}\right)\right), \tag{3}$$

and thus, for large values of n, we can use the approximation

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n. \tag{4}$$

D. Gaussian Approximation

Another tool which we will frequently use is the following Gaussian approximation technique. We consider the symbols x_i in a sequence $\mathbf{x}$ as n independent random variables which are uniformly drawn from the alphabet $\mathcal{A}_q$. We are interested in the distribution of the sum $\sum_{i=1}^n \phi(x_i)$, where ϕ is a function mapping symbols from $\mathcal{A}_q$ to real numbers, which has the property that the possible outcomes of the sum form a set of consecutive integer numbers. Then, by the Central Limit Theorem, the probability that this sum takes the integer value s is approximately

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{s-\mu}{\sigma})^2},$$

with mean

$$\mu = nE[\phi(x)] = \frac{n}{q}\sum_{j=0}^{q-1}\phi(q-1-2j) \tag{5}$$

and variance

$$\begin{aligned}\sigma^2 &= n(E[(\phi(x))^2] - (E[\phi(x)])^2) \\ &= n\left(\left(\frac{1}{q}\sum_{j=0}^{q-1}(\phi(q-1-2j))^2\right) - \left(\frac{\mu}{n}\right)^2\right).\end{aligned} \tag{6}$$

Hence, the number of q-ary sequences of length n with $\sum_{i=1}^n \phi(x_i)$ equal to s is approximately

$$q^n \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{s-\mu}{\sigma})^2}. \tag{7}$$

Note that for fixed n and q this expression is maximum if s is equal to μ, which leads to a minimum redundancy of

$$\log_q \sigma + \frac{1}{2}\log_q 2\pi$$

when substituting (7) for M in (2).

III. Minimum Redundancy of Balanced Codes

In this section, we consider the cardinalities of q-ary SB, CB, PB, and CPB codes. From these cardinalities we derive asymptotic expressions for the minimum redundancies. The SB and CB results have been known for a long time but are reconsidered here for completeness. The PB and CPB results are new.

A. Symbol-Balanced Sequences

For an SB code, all q alphabet symbols must appear equally often in any codeword of length n. Hence, the problem of determining the number of such words boils down to a standard combinatorial problem. This number and the consequence with respect to minimum redundancy, as already discussed in [11], are as follows.

Theorem 1. *For any q and $n = mq$, it holds that*

$$\begin{aligned}M_{\mathrm{SB}}(n,q) &= \frac{n!}{((n/q)!)^q} \\ &\approx q^n\left(\frac{1}{2\pi n}\right)^{\frac{q-1}{2}} q^{\frac{q}{2}}.\end{aligned}$$

Proof. The equality follows from straightforward combinatorics and the approximation from multiple uses of Stirling's formula (4). □

Corollary 2. *For any q and $n = mq$, it holds that*

$$\begin{aligned}r_{\mathrm{SB}}(n,q) &= n - \log_q M_{\mathrm{SB}}(n,q) \\ &\approx \frac{q-1}{2}\log_q n + \frac{q-1}{2}\log_q 2\pi - \frac{q}{2}.\end{aligned}$$

Proof. The equality follows (by definition) from (2) and the approximation from Theorem 1. □

By using (3) rather than (4), the more precise expressions

$$M_{\mathrm{SB}}(n,q) = q^n\left(\frac{1}{2\pi n}\right)^{\frac{q-1}{2}} q^{\frac{q}{2}}\left(1 + O\left(\frac{1}{n}\right)\right)$$

and

$$r_{\mathrm{SB}}(n,q) = \frac{q-1}{2}\log_q n + \frac{q-1}{2}\log_q 2\pi - \frac{q}{2} + O\left(\frac{1}{n}\right)$$

are obtained. Hence, the approximation from Corollary 2 is exact if $n \to \infty$. This also holds for the approximate minimum redundancy expressions which will be presented in the subsequent subsections. In Subsection III-E, we will illustrate the accuracy of the approximate expressions for finite values of n.

B. Charge-Balanced Sequences

As observed by Capocelli *et al.* [6] in their investigation of q-ary immutable codes, the number of words in a CB code of length n was studied by Star [15] in the context of his analysis of the number of restricted compositions of a positive integer. The final result is as stated in the next theorem, for which we provide a simple alternative proof.

Theorem 3. *For any q and n (which is even if q is even), it holds that*

$$M_{\mathrm{CB}}(n,q) \approx q^n\sqrt{\frac{6}{\pi n(q^2-1)}}.$$

Proof. We use the Gaussian approximation technique as discussed in Subsection II-D. Choosing the function ϕ to be

$$\phi(x) = \frac{x}{2}, \tag{8}$$

it follows that the number of sequences $\mathbf{x}$ over $\mathcal{A}_q$ of length n with $\sum_{i=1}^{n} x_i = s$ is approximately equal to (7) with mean

$$\mu = \frac{n}{q} \sum_{j=0}^{q-1} \frac{q-1-2j}{2} = 0 \tag{9}$$

(from (5) and (8)) and variance

$$\sigma^2 = n \left(\frac{1}{q} \sum_{j=0}^{q-1} \left(\frac{q-1-2j}{2} \right)^2 \right) = n \frac{q^2-1}{12} \tag{10}$$

(from (6), (8), and (9)). Note that CB sequences are characterized by the fact that $s = 0$, and thus substitution of this value in (7), with $\mu = 0$ and $\sigma^2 = n(q^2-1)/12$, provides an approximation of $M_{\mathrm{CB}}(n, q)$. The result is as given in the theorem. □

Corollary 4. *For any q and n (which is even if q is even), it holds that*

$$\begin{aligned} r_{\mathrm{CB}}(n,q) &= n - \log_q M_{\mathrm{CB}}(n,q) \\ &\approx \frac{1}{2}\log_q n + \frac{1}{2}\log_q \frac{\pi(q^2-1)}{6}. \end{aligned}$$

Proof. The equality follows (by definition) from (2) and the approximation from Theorem 3. □

C. Polarity-Balanced Sequences

When calculating the number of q-ary PB sequences of length n, we distinguish between the cases q is even and q is odd, since in the latter case we should take into account the fact that the code alphabet contains the symbol '0' which is of indeterminate polarity. The results are presented in the next theorems, while expressions for the minimum redundancies of PB codes are given in the subsequent corollaries.

Theorem 5. *For any even q and even n, it holds that*

$$\begin{aligned} M_{\mathrm{PB}}(n,q) &= \binom{n}{n/2} \left(\frac{q}{2}\right)^n \qquad (11) \\ &\approx q^n \sqrt{\frac{2}{\pi n}}. \qquad (12) \end{aligned}$$

Proof. The equality (11) follows by observing that there are $\binom{n}{n/2}$ ways to create a balanced polarity pattern over n positions and that for each such pattern we have $q/2$ symbol options for every positions. The approximation can be obtained by multiple uses of Stirling's formula (4) or by applying the Gaussian approximation technique discussed in Subsection II-D. Here, we opt for the latter, since intermediate results also turn out to be useful for the CPB case. Choosing the function ϕ to be

$$\phi(x) = \begin{cases} -\frac{1}{2}, & \text{if } x < 0, \\ +\frac{1}{2}, & \text{if } x > 0, \end{cases} \tag{13}$$

it follows that the number of q-ary sequences $\mathbf{x}$ of length n with $\sum_{i=1}^{n} \phi(x_i) = s$ is approximately equal to (7) with mean

$$\mu = \frac{n}{q} \sum_{j=0}^{q-1} \phi(q-1-2j) = 0 \tag{14}$$

(from (5) and (13)) and variance

$$\sigma^2 = n \left(\frac{1}{q} \sum_{j=0}^{q-1} (\phi(q-1-2j))^2 \right) = \frac{n}{4} \tag{15}$$

(from (6), (13), and (14)). Note that PB sequences are characterized by the fact that $s = 0$, and thus substitution of this value in (7), with $\mu = 0$ and $\sigma^2 = n/4$, gives (12). □

Corollary 6. *For any even q and even n, it holds that*

$$\begin{aligned} r_{\mathrm{PB}}(n,q) &= n - \log_q M_{\mathrm{PB}}(n,q) \\ &\approx \frac{1}{2}\log_q n + \frac{1}{2}\log_q \frac{\pi}{2}. \end{aligned}$$

Proof. The equality follows (by definition) from (2) and the approximation from Theorem 5. □

Theorem 7. *For any n and odd q, it holds that*

$$\begin{aligned} M_{\mathrm{PB}}(n,q) &= \sum_{j=0}^{\lfloor n/2 \rfloor} \frac{n!}{j!j!(n-2j)!} \left(\frac{q-1}{2}\right)^{2j} \qquad (16) \\ &\approx q^n \sqrt{\frac{q}{2\pi n(q-1)}}. \qquad (17) \end{aligned}$$

Proof. The number of q-ary PB sequences of length n with j positive symbols, j negative symbols, and thus $n-2j$ neutral symbols, is $\frac{n!}{j!j!(n-2j)!}\left(\frac{q-1}{2}\right)^{2j}$, since there are $\frac{n!}{j!j!(n-2j)!}$ ways to create the postive/negative/neutral pattern over n positions and for each such pattern we have $(q-1)/2$ symbol options for every non-neutral position. Summing over all possible values of j shows (16).

In order to obtain a simple expression for large values of n, we again use the Gaussian approximation technique introduced in Subsection II-D. Proceeding as in the proof of Theorem 5, while replacing the function ϕ by

$$\phi(x) = \begin{cases} -1, & \text{if } x < 0, \\ 0, & \text{if } x = 0, \\ +1, & \text{if } x > 0, \end{cases} \tag{18}$$

giving mean

$$\mu = \frac{n}{q} \sum_{j=0}^{q-1} \phi(q-1-2j) = 0 \tag{19}$$

(from (5) and (18)) and variance

$$\sigma^2 = n \left(\frac{1}{q} \sum_{j=0}^{q-1} (\phi(q-1-2j))^2 \right) = \frac{n(q-1)}{q} \tag{20}$$

(from (6), (18) and (19)), we obtain (17). □

Corollary 8. *For any n and odd q, it holds that*

$$\begin{aligned} r_{\mathrm{PB}}(n,q) &= n - \log_q M_{\mathrm{PB}}(n,q) \\ &\approx \frac{1}{2}\log_q n + \frac{1}{2}\log_q \frac{2\pi(q-1)}{q}. \end{aligned}$$

Proof. The equality follows (by definition) from (2) and the approximation from Theorem 7. □

D. Charge & Polarity-Balanced Sequences

Since each of the alphabets $\mathcal{A}_2 = \{-1,+1\}$ and $\mathcal{A}_3 = \{-2,0,+2\}$ has exactly one positive and one negative symbol, which have equal absolute value, it follows immediately from the definitions that the CB and PB constraints are completely equivalent for sequences over these alphabets. Therefore, for $q \leqslant 3$, any CB sequence is also PB, and vice versa.

Hence, the minimum redundancy of a binary/bipolar CPB code of even length n satisfies

$$\begin{aligned} r_{\mathrm{CPB}}(n,2) &= r_{\mathrm{CB}}(n,2) = r_{\mathrm{PB}}(n,2) \\ &\approx \frac{1}{2}\log_2 n + \frac{1}{2}\log_2 \frac{\pi}{2}, \end{aligned}$$

where the final expression follows from Corollary 4 or 6. Furthermore, note that we have the same expression for $r_{\mathrm{SB}}(n,2)$; see Corollary 2. This does not come as a surprise, as all balancing perspectives under consideration in the paper are equivalent in the binary/bipolar case.

For the minimum redundancy of a ternary CPB code of length n we find

$$\begin{aligned} r_{\mathrm{CPB}}(n,3) &= r_{\mathrm{CB}}(n,3) = r_{\mathrm{PB}}(n,3) \\ &\approx \frac{1}{2}\log_3 n + \frac{1}{2}\log_3 \frac{4\pi}{3}, \end{aligned}$$

where the final expression follows from Corollary 4 or 8. In this case, the corresponding expression for symbol balancing, provided by Corollary 2, is

$$r_{\mathrm{SB}}(n,3) \approx \log_3 n + \log_3 2\pi - \frac{3}{2},$$

which exceeds $r_{\mathrm{CPB}}(n,3)$ roughly by a factor of two.

As already argued in Section I, the notions of CB and PB are not the same in case $q \geqslant 4$. First, we precisely determine, by combinatorial arguments, the number of CPB sequences of length n in case $q = 4$. Then, we derive approximate expressions for the number of CPB sequences for $q \geqslant 4$, from which we obtain the minimum redundancy.

We can count the number of CPB sequences over $\mathcal{A}_4$ of even length n as follows. Polarity balancing requires that $n/2$ positions take values in $\{-3,-1\}$. If the number of such positions taking value -3 is i, then charge balancing requires that in the complementary set of $n/2$ positions taking values in $\{+1,+3\}$ there must be i positions that take the value $+3$. Therefore, the size of the intersection of the sets of CB and PB sequences is given by

$$\begin{aligned} M_{\mathrm{CPB}}(n,4) &= \binom{n}{n/2}\left(\sum_{i=0}^{n/2}\binom{n/2}{i}\binom{n/2}{i}\right) \\ &= \binom{n}{n/2}\left(\sum_{i=0}^{n/2}\binom{n/2}{i}\binom{n/2}{(n/2)-i}\right) \\ &= \binom{n}{n/2}\binom{n}{n/2} = \binom{n}{n/2}^2. \end{aligned} \tag{21}$$

It seems to be cumbersome to extend the arguments used in the $q = 4$ case to determine $M_{\mathrm{CPB}}(n,q)$ for larger values of q. However, the elegant Gaussian approximation method is still feasible, albeit that we need a joint distribution this time, since we have two constraints. The results are presented in the next theorems and corollaries.

Theorem 9. *For any even $q \geqslant 4$ and even n, it holds that*

$$M_{\mathrm{CPB}}(n,q) \approx q^n \frac{1}{\pi n}\sqrt{\frac{48}{q^2-4}}.$$

Proof. We consider the symbols x_i in a sequence $\mathbf{x}$ as n independent random variables which are uniformly drawn from the alphabet $\mathcal{A}_q$ with $q \geqslant 4$ even. We are interested in the joint distribution of the sums $S_1 = \sum_{i=1}^{n} x_i/2$ and $S_2 = \sum_{i=1}^{n}\phi(x_i)$, where ϕ is as defined in (13). The probability that these sums take the integer values s_1 and s_2, respectively, is approximately

$$\frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}f(s_1,s_2)},$$

where

$$f(s_1,s_2) = \sum_{i=1}^{2}\left(\frac{s_i-\mu_i}{\sigma_i}\right)^2 - \frac{2\rho(s_1-\mu_1)(s_2-\mu_2)}{\sigma_1\sigma_2},$$

$$\mu_1 = 0 \text{ (from (9))},$$

$$\sigma_1 = \sqrt{\frac{n(q^2-1)}{12}} \text{ (from (10))},$$

$$\mu_2 = 0 \text{ (from (14))},$$

$$\sigma_2 = \sqrt{\frac{n}{4}} \text{ (from (15))},$$

and the correlation coefficient is

$$\begin{aligned} \rho &= \frac{E[(S_1-\mu_1)(S_2-\mu_2)]}{\sigma_1\sigma_2} = \frac{E[S_1S_2]}{\sqrt{\frac{n(q^2-1)}{12}}\sqrt{\frac{n}{4}}} \\ &= \frac{\frac{n}{2q}\sum_{i=0}^{\frac{q}{2}-1}(q-1-2i)}{n\sqrt{\frac{(q^2-1)}{48}}} = \sqrt{\frac{3q^2}{4(q^2-1)}}. \end{aligned}$$

Hence, the number of q-ary sequences of length n with $S_1 = s_1$ and $S_2 = s_2$ is approximately

$$q^n \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}f(s_1,s_2)}. \tag{22}$$

Substitution of $s_1 = 0$ (the charge constraint), $s_2 = 0$ (the polarity constraint), and the two mean values, the two standard deviations, and the correlation coefficient, gives the stated result. □

Note that this theorem gives

$$M_{\mathrm{CPB}}(n,4) \approx 4^n \frac{2}{\pi n},$$

a result which can also be obtained by applying the Stirling formula (4) multiple times on (21).

Corollary 10. *For any even $q \geqslant 4$ and even n, it holds that*

$$\begin{aligned} r_{\mathrm{CPB}}(n,q) &= n - \log_q M_{\mathrm{CPB}}(n,q) \\ &\approx \log_q n + \log_q\left(\pi\sqrt{\frac{q^2-4}{48}}\right). \end{aligned}$$

Proof. The equality follows (by definition) from (2) and the approximation from Theorem 9. □

Theorem 11. *For any n and odd $q \geqslant 5$, it holds that*

$$M_{\mathrm{CPB}}(n,q) \approx q^n \frac{1}{\pi n}\sqrt{\frac{12q^2}{(q^2-1)(q-1)(q-3)}}.$$

Proof. We follow the same reasoning as in the proof of Theorem 9, though now using (18) instead of (13) for the ϕ function. Consequently, the standard deviation of S_2 changes to

$$\sigma_2 = \sqrt{\frac{n(q-1)}{q}} \text{ (from (20))},$$

and the correlation coefficient to

$$\begin{aligned}\rho &= \frac{E[(S_1-\mu_1)(S_2-\mu_2)]}{\sigma_1\sigma_2} = \frac{E[S_1S_2]}{\sqrt{\frac{n(q^2-1)}{12}}\sqrt{\frac{n(q-1)}{q}}} \\ &= \frac{\frac{n}{2q}\sum_{i=0}^{\frac{q-3}{2}}(q-1-2i)}{n\sqrt{\frac{(q^2-1)(q-1)}{12q}}} = \sqrt{\frac{3(q+1)}{4q}}.\end{aligned}$$

The final result follows by substituting all the parameters in (22). □

Corollary 12. *For any n and odd $q \geqslant 5$, it holds that*

$$\begin{aligned} r_{\mathrm{CPB}}(n,q) &= n - \log_q M_{\mathrm{CPB}}(n,q) \\ &\approx \log_q n + \\ &\quad \log_q\left(\pi\sqrt{\frac{(q^2-1)(q-1)(q-3)}{12q^2}}\right).\end{aligned}$$

Proof. The equality follows (by definition) from (2) and the approximation from Theorem 11. □

E. Discussion

In this subsection, we discuss the results on the minimum redundancy of balanced codes as obtained in this section. As stated before, the minimum redundancy expressions as presented in the corollaries are approximations which are exact if $n \to \infty$. For finite values of n, the accuracy of these expressions depends on the convergence rates of the underlying Stirling/Gaussian approximations. Here, we provide an illustration by showing some numerical values for $r_{\mathrm{CPB}}(n,4)$, i.e., the minimum redundancy of a CPB code of length n over $\mathcal{A}_4$. From (2) and (21) we obtain the exact expression

$$r_{\mathrm{CPB}}(n,4) = n - 2\log_4\binom{n}{n/2}, \tag{23}$$

while Corollary 10 gives the approximate expression

$$r_{\mathrm{CPB}}(n,4) \approx \log_4(n\pi/2). \tag{24}$$

The comparison of these two expressions as given in Table I shows that the approximation is quite accurate, even for small values of n.

TABLE I
NUMERICAL VALUES FOR $r_{\mathrm{CPB}}(n,4)$

n	Exact, Eq. (23)	Approximation, Eq. (24)
10	2.0227	1.9867
20	2.5047	2.4867
40	2.9957	2.9867
60	3.2852	3.2792
80	3.4912	3.4867
100	3.6513	3.6477
200	4.1495	4.1477
400	4.6486	4.6477
600	4.9408	4.9402
800	5.1481	5.1477
1000	5.3090	5.3086

TABLE II
ASYMPTOTIC NORMALIZED REDUNDANCIES

	SB	CB	PB	CPB
$q=2$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
$q=3$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
$q \geqslant 4$	$\frac{q-1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1

Note that all minimum redundancy expressions are of the form

$$g(q)\log_q n + h(q),$$

where g and h are functions such that the output values may depend on the alphabet size q but not on the block length n. For comparison purposes, we introduce the *asymptotic normalized redundancy* (ANR) as the redundancy divided by $\log_q n$ in the limit of large values of n. Note that this ANR is equal to $g(q)$. For example, it follows from Corollary 2 that

$$g_{\mathrm{SB}}(q) = \frac{q-1}{2}.$$

The complete overview of these ANRs is provided in Table II. From this table, we conclude that the CB and PB properties are equally expensive in terms of ANR, while the SB property is $q-1$ times as expensive. The combined CB and PB property (CPB) is as expensive as either of the individual properties, i.e., the other comes for free, if $q \leqslant 3$, while it costs the sum of the individual contributions if $q \geqslant 4$.

IV. CONSTRUCTIONS OF BALANCED CODES

In the previous section we have determined expressions for the number $M(n,q)$ of q-ary sequences of length n satisfying certain balancing constraints. From these expressions we calculated the minimum required code redundancy to achieve the constraints. However, the lists of balanced words come with little structure. Applying table look-up is only feasible for small codes, but for practical implementation of larger codes, we need simple encoding and decoding algorithms. Knuth presented such an algorithm for the case $q = 2$, i.e., for binary/bipolar balanced codes [9]. Here, we will propose extensions to non-binary codes from various balancing perspectives.

All proposed methods take an approach similar to the original Knuth construction. We make simple and reversible modifications to a q-ary information sequence $\mathbf{u}$ of length k to obtain a q-ary balanced sequence $\mathbf{x}$ of the same length. Next, we create a q-ary balanced prefix $\mathbf{p}$ of length p, which uniquely identifies the modifications. The q-ary balanced codeword $\mathbf{c} = (\mathbf{p}, \mathbf{x})$ of length $n = p + k$ is then transmitted or stored. The receiver retrieves the modifications from the prefix and applies these in reverse on $\mathbf{x}$ to obtain the original $\mathbf{u}$.

The constructions are nice and simple, but not optimal with respect to redundancy. Note that all codewords consist of two parts which are both balanced, and thus words which are balanced overall, but not within these parts, are excluded. Hence, simplicity comes at a price of increased redundancy. In order to still keep the redundancy as small as possible within the construction framework, we should minimize the prefix length p. Since the prefix is much shorter than the information sequence, we will assume that encoding and decoding of the prefix can be done by table look-up or another minimum redundancy achieving method. Let the number of different prefixes required to uniquely identify the modifications be denoted by P. Ignoring balancing, the number of q-ary symbols needed to represent the prefix is thus

$$p' = \log_q P, \tag{25}$$

which we will call the *unbalanced redundancy*. The actual prefix length will be (a little bit) larger, since the prefix needs to be balanced as well. It should be chosen as the smallest integer p such that

$$M(p, q) \geqslant P. \tag{26}$$

The analysis from the previous section shows that, for fixed q, the extra redundancy to make the prefix balanced is in the order of $\log p'$, i.e.,

$$p = p' + O(\log p').$$

Hence, for rough evaluation purposes, the unbalanced redundancy p', which is easily determined by (25), may serve as a satisfactory approximation of the actual redundancy p, which requires the more cumbersome computation from (26).

All constructions will be presented for the code alphabet $\mathcal{A}_q$, but equivalents for other alphabets, e.g., $\mathbb{Z}_q$, can be established using the mapping (1). Before starting the descriptions of the constructions, we introduce some more notation. The real sum of all symbols in a q-ary sequence $\mathbf{y}$ is denoted by $\text{Sum}(\mathbf{y})$, i.e.,

$$\text{Sum}(\mathbf{y}) = \sum_i y_i.$$

Further, let $S_j(\mathbf{y})$ denote the number of appearances of the alphabet symbol j in $\mathbf{y}$, i.e.,

$$S_j(\mathbf{y}) = |\{i : y_i = j\}$$

for any alphabet symbol j. Finally, as a short-hand notation, we denote a run of b symbols a by a^b, e.g., $3^2 1^3 (-1)^1 3^2$ denotes the sequence $(3, 3, 1, 1, 1, -1, 3, 3)$.

A. Knuth's Construction

We start by stating Knuth's original construction for bipolar codes [9], as a reference. For any information sequence $\mathbf{u}$ of even length k and any $j \in \{0, 1, \ldots, k\}$, let $\mathbf{u}'_j$ denote the sequence $\mathbf{u}$ with the first j symbols multiplied by -1. A *balancing index* is a number z for which $\mathbf{u}'_z$ is balanced.

Knuth Encoding Procedure

1) Determine a balancing index $z \in \{0, 1, \ldots, k-1\}$ for the information sequence $\mathbf{u}$.
2) Multiply the first z symbols of $\mathbf{u}$ by -1 to obtain the balanced sequence $\mathbf{x}$.
3) Map z to a unique balanced prefix $\mathbf{p}$.

Then transmit or store the balanced codeword $\mathbf{c} = (\mathbf{p}, \mathbf{x})$.

Knuth Decoding Procedure

1) Retrieve the balancing index z from $\mathbf{p}$.
2) Multiply the first z symbols of $\mathbf{x}$ by -1 to retrieve $\mathbf{u}$.

Proof. It is easy to see that the operation in the encoding procedure is properly reversed in the decoding procedure. Hence, we only need to show that for every sequence $\mathbf{u}$ of length k there exists at least one $z \in \{0, 1, \ldots, k-1\}$ such that $\mathbf{u}'_z$ is balanced, i.e., $\text{Sum}(\mathbf{u}'_z) = 0$. This immediately follows from combining the following observations.

1) $\text{Sum}(\mathbf{u}'_0)$ is even.
2) $\text{Sum}(\mathbf{u}'_j) = \text{Sum}(\mathbf{u}'_{j-1}) \pm 2$ for all $j \in \{1, 2, \ldots, k\}$.
3) $\text{Sum}(\mathbf{u}'_k) = -\text{Sum}(\mathbf{u}'_0)$.

□

Since there are k possible values for z, the redundancy, i.e., the length p of the prefix, is a little bit more than $p' = \log_2 k$.

Example 1. For the bipolar sequence

$$\mathbf{u} = (+1, -1, +1, +1, +1, +1)$$

of length 6, encoding goes as follows.

1) Find the balancing index to be $z = 4$.
2) Invert the first 4 positions of $\mathbf{u}$, i.e.,

$$\mathbf{x} = (-1, +1, -1, -1, +1, +1).$$

3) Uniquely map the balancing index 4 to one of the six balanced sequences of length four, e.g.,

$$\mathbf{p} = (+1, -1, -1, +1).$$

Then the balanced transmitted/stored sequence is

$$\mathbf{c} = (\mathbf{p}, \mathbf{x}) = (+1, -1, -1, +1, -1, +1, -1, -1, +1, +1).$$

B. Polarity-Balanced Code Construction

Knuth's original method for generating balanced binary sequences can be adapted to generate q-ary PB sequences. This is rather straightforward, although there is a snag if q is odd. In this case, the number of zero-valued symbols in $\mathbf{u}$ may be of different parity than the length k, which results in an odd number of non-zero (either positive or negative) symbols. Since the value zero is (polarity-)neutral, i.e., neither positive nor negative, inversion of any number of symbols in $\mathbf{u}$ will not lead to a PB sequence in such a situation. We will solve this by introducing an offset in case q is odd. We propose the following algorithm for sequences over $\mathcal{A}_q$, where $\oplus_{2q}$ denotes the addition over the integer numbers, with a reduction modulo $2q$ such that the final outcome is in $\mathcal{A}_q$.

PB Encoding Procedure

1) If q is odd, then determine a symbol a in $\mathcal{A}_q$ such that $S_a(\mathbf{u})$ has the same parity as the length k of $\mathbf{u}$, i.e., $S_a(\mathbf{u})$ and k are either both even or both odd.
2) If q is odd, then compute $\mathbf{u}' = \mathbf{u} \oplus_{2q} (-\mathbf{a})$, where $\mathbf{a} = (a, a, \dots, a)$ is of length k. If q is even, then $\mathbf{u}' = \mathbf{u}$.
3) Determine a polarity balancing index $z \in \{0, 1, \dots, k-1\}$ for $\mathbf{u}'$.
4) Multiply the first z positions of $\mathbf{u}'$ by -1 to obtain the PB sequence $\mathbf{x}$.
5) Map z (if q is even) or (a, z) (if q is odd) to a unique PB prefix $\mathbf{p}$.

Then transmit or store the balanced codeword $\mathbf{c} = (\mathbf{p}, \mathbf{x})$.

PB Decoding Procedure

1) Retrieve the balancing index z from $\mathbf{p}$.
2) Multiply the first z positions of $\mathbf{x}$ by -1 to retrieve $\mathbf{u}$ (if q is even) or $\mathbf{u}'$ (if q is odd).
3) If q is odd, then retrieve a from the prefix $\mathbf{p}$ and compute $\mathbf{u} = \mathbf{u}' \oplus_{2q} \mathbf{a}$.

Proof. It is easy to see that the operations in the encoding procedure are properly reversed in the decoding procedure. Hence, we only need to show the existence of (i) a suitable offset a (in case q odd) and (ii) a suitable polarity balancing index z.

(i) The existence of a can be demonstrated by supposing it does not exist and then deriving a contradiction. If q and k are odd, then $S_j(\mathbf{u})$ is odd for at least one symbol $j \in \mathcal{A}_q$, since all of them being even would imply that $k = \sum_i S_i(\mathbf{u})$ is even. If q is odd and k is even, then $S_j(\mathbf{u})$ is even for at least one $j \in \mathcal{A}_q$, since all of them being odd would imply that $k = \sum_i S_i(\mathbf{u})$, a summation of an odd number of odd terms, is odd.

(ii) The existence of z follows by a similar argument as for the Knuth algorithm. Let $\mathbf{u}'_j$ denote the sequence $\mathbf{u}'$ with the first j symbols multiplied by -1 and let ϕ be defined as in (18). For a PB balancing index z, it must hold that $\mathrm{Sum}(\phi(\mathbf{u}'_j)) = 0$. The existence of a PB balancing index follows by combining the following observations.

1) $\mathrm{Sum}(\phi(\mathbf{u}'_0))$ is even, since the number of non-zero symbols in $\mathbf{u}'$ is even.
2) $\mathrm{Sum}(\phi(\mathbf{u}'_j)) = \mathrm{Sum}(\phi(\mathbf{u}'_{j-1})) + c$ for all $j \in \{1, 2, \dots, k\}$, where $c \in \{-2, 0, +2\}$.
3) $\mathrm{Sum}(\phi(\mathbf{u}'_k)) = -\mathrm{Sum}(\phi(\mathbf{u}'_0))$.

□

Since there are k possible values for z and q possible values for a, we have $p' = \log_q k$ if q is even and $p' = 1 + \log_q k$ if q is odd.

Example 2. Let $q = 5$. For the sequence

$$\mathbf{u} = (+4, +4, -2, 0, 0, 0, 0) \in (\mathcal{A}_5)^7,$$

encoding goes as follows.

1) Since $q = 5$ and $k = 7$ are odd, identify '-2' as the symbol a with an odd number of appearances in $\mathbf{u}$.
2) Subtract (modulo 10) the value -2 from every symbol in $\mathbf{u}$, resulting in
$$\mathbf{u}' = (-4, -4, 0, +2, +2, +2, +2).$$
3) Find the PB index z to be 6.
4) Multiply the first 6 positions of $\mathbf{u}'$ by -1 to obtain
$$\mathbf{x} = (+4, +4, 0, -2, -2, -2, +2).$$
5) Uniquely map $(a, z) = (-2, 6)$ to one of the PB sequences of length 4, e.g.,
$$\mathbf{p} = (+2, 0, 0, -4).$$

Then the balanced transmitted/stored sequence is

$$\mathbf{c} = (+2, 0, 0, -4, +4, +4, 0, -2, -2, -2, +2).$$

C. Charge-Balanced Code Construction

In [16], Swart and Weber presented a Knuth-like construction for q-ary CB codes over the alphabet $\mathbb{Z}_q$. We include it here, in a version for the alphabet $\mathcal{A}_q$, to make this paper self-contained. Furthermore, we need it in the subsequent subsection as a component for CPB code construction. The key ingredient of the CB method is a set of qk balancing sequences $\mathbf{b}_i$, $i = 0, 1, \dots, qk-1$, each consisting of g symbols $j+2$ followed by $k-g$ symbols j, i.e.,

$$\mathbf{b}_i = (j+2)^g j^{k-g},$$

where $j = 2\lfloor i/k \rfloor$ and $g = i - k\lfloor i/k \rfloor$. Again, $\oplus_{2q}$ denotes the addition over the integer numbers, with a reduction modulo $2q$ such that the final outcome is in $\mathcal{A}_q$. A charge balancing index is a number z such that $\mathrm{Sum}(\mathbf{u} \oplus_{2q} \mathbf{b}_z) = 0$. The algorithm is described as follows.

CB Encoding Procedure

1) Determine a CB index $z \in \{0, 1, \dots, qk-1\}$ for the information sequence $\mathbf{u}$.
2) Compute the CB sequence $\mathbf{x} = \mathbf{u} \oplus_{2q} \mathbf{b}_z$.
3) Map z to a unique CB prefix $\mathbf{p}$.

Then transmit or store the balanced codeword $\mathbf{c} = (\mathbf{p}, \mathbf{x})$.

CB Decoding Procedure

1) Retrieve the balancing index z from $\mathbf{p}$.
2) Compute $\mathbf{u} = \mathbf{x} \oplus_{2q} (-\mathbf{b}_z)$.

Proof. It is easy to see that the operation in the encoding procedure is properly reversed in the decoding procedure. Hence, we only need to show the existence of a CB index for any information sequence $\mathbf{u}$ of length k. Define $\mathbf{b}_{qk} = \mathbf{b}_0$, and consider the series

$$\mathrm{Sum}(\mathbf{u} \oplus_{2q} \mathbf{b}_0), \mathrm{Sum}(\mathbf{u} \oplus_{2q} \mathbf{b}_1), \dots, \mathrm{Sum}(\mathbf{u} \oplus_{2q} \mathbf{b}_{qk}).$$

We make the following observations.

1) The series starts and ends with the same even value.
2) For all $i \in \{0, 1, \dots, qk-1\}$, it holds that
$$\mathrm{Sum}(\mathbf{u} \oplus_{2q} \mathbf{b}_{i+1}) = \mathrm{Sum}(\mathbf{u} \oplus_{2q} \mathbf{b}_i) + c,$$
where c is either $+2$ or $-2q+2$.

3) It holds that

$$\begin{aligned}\sum_{j=0}^{q-1}\mathrm{Sum}(\mathbf{u}\oplus_{2q}\mathbf{b}_{jk}) &= \sum_{l=1}^{k}\sum_{j=0}^{q-1}(u_l\oplus_{2q}2j)\\ &= k\sum_{j=0}^{q-1}(-q+1+2j)=0,\end{aligned}$$

where the first equality follows from the fact that the sequence $\mathbf{b}_{jk}$ consists of k symbols $2j$, and the second equality from the consequence that every position l takes every symbol value from the alphabet $\mathcal{A}_q$ exactly once in the summation. Hence, the average value of all $\mathrm{Sum}(\mathbf{u}\oplus_{2q}\mathbf{b}_{jk})$, with $j=0,1,\ldots,q-1$, is 0.

By combining these three observations, we can conclude that there exists at least one z in $\{0,1,\ldots,qk-1\}$ such that $\mathrm{Sum}(\mathbf{u}\oplus_{2q}\mathbf{b}_z)=0$. □

Since there are qk possible values for z, the unbalanced redundancy is $p'=1+\log_q k$. Note that by setting $q=2$, we do not exactly get the original Knuth method as described in Subsection IV-A, where p' is one bit less. The reason is that for the binary case, it can be shown (as done by Knuth and in Subsection IV-A) that there is always a suitable balancing index in a set of k candidates (rather than $2k$). For further details, see [16]. Pelusi *et al.* [14] presented a slightly improved q-ary CB coding scheme, using $(q-1)k+q \bmod 2$ rather than qk balancing functions, with the same asymptotic redundancy though.

Example 3. We use the same information sequence as in Example 2, i.e.,

$$\mathbf{u}=(+4,+4,-2,0,0,0,0)\in(\mathcal{A}_5)^7.$$

Encoding into a CB sequence goes as follows.

1) Find a suitable CB index z to be 32.
2) Compute the CB sequence

$$\begin{aligned}\mathbf{x} &= \mathbf{u}\oplus_{10}(\mathbf{b}_{32})\\ &= (+4,+4,-2,0,0,0,0)\oplus_{10}\\ &\quad\ (10,10,10,10,8,8,8)\\ &= (+4,+4,-2,0,-2,-2,-2)\end{aligned}$$

3) Uniquely map the CB index 32 to one of the CB sequences of length 4, e.g.,

$$\mathbf{p}=(+4,0,-2,-2).$$

Then the balanced transmitted/stored sequence is

$$\mathbf{c}=(+4,0,-2,-2,+4,+4,-2,0,-2,-2,-2).$$

Note that the sequence $\mathbf{x}$ generated this way is not PB. Rather than $z=32$, we could also have chosen $z=7$, but also then the resulting CB sequence

$$\mathbf{x}=(-4,-4,0,+2,+2,+2,+2)$$

is not PB.

D. Charge & Polarity-Balanced Code Construction

If $q\leqslant 3$, then any code which is PB is also CB and vice versa. Hence, either of the coding strategies described in the previous two subsections provides CPB codes. However, for $q\geqslant 4$, the CB and PB properties are no longer equivalent, and a dedicated construction method is needed. Such a method will be proposed in this subsection, where we will assume throughout that $q\geqslant 4$ and that k is even if q even.

For constructing codes having both the charge and polarity balancing properties, we can still base our constructions on the methods described in the previous two subsections. However, the straightforward strategy of first applying one method and then the other could fail, since the property obtained in the first round might be destroyed in the second. Therefore, a more sophisticated strategy should be developed.

In the proposed method, we first transform the information sequence $\mathbf{u}$ into a PB sequence as described in Subsection IV-B. In this PB sequence, which we denote by $\mathbf{y}$, we focus on the subsequences $\mathbf{y}^+$, which consists of all positive symbols in $\mathbf{y}$, and $\mathbf{y}^-$, which consists of all negative symbols. Both subsequences have the same length (due to the established PB property) which we denote by k'. Note that

$$\mathrm{Sum}(\mathbf{y}^-)\leqslant 0\leqslant \mathrm{Sum}(\mathbf{y}^+).$$

We are going to make modifications to $\mathbf{y}$, affecting only $\mathbf{y}^+$ and $\mathbf{y}^-$, such that the resulting sequence $\mathbf{x}$ satisfies

$$\mathrm{Sum}(\mathbf{x}^+)+\mathrm{Sum}(\mathbf{x}^-)=0, \qquad (27)$$

which implies that $\mathbf{x}$ is CPB.

The modifications are done in such a way that the polarity of all involved symbols will not change. Hence, like $\mathbf{y}$, the sequence $\mathbf{x}$ is PB. The first step of the modification process consists of a possible 'mirror' operation on the symbols in $\mathbf{y}^+$ (with respect to the value $\lceil q/2\rceil$). Define

$$\xi=\begin{cases}1, & \text{if } \mathrm{Sum}(\mathbf{y}^+)<k'\lceil q/2\rceil<-\mathrm{Sum}(\mathbf{y}^-)\\ & \text{or } -\mathrm{Sum}(\mathbf{y}^-)<k'\lceil q/2\rceil<\mathrm{Sum}(\mathbf{y}^+),\\ 0, & \text{otherwise.}\end{cases} \qquad (28)$$

If $\xi=1$, then all symbols y_i in $\mathbf{y}^+$ are replaced by $2\lceil q/2\rceil-y_i$; else they are left untouched. Note that for the sequence $\mathbf{z}$ obtained from $\mathbf{y}$ by this operation, it holds that $\mathrm{Sum}(\mathbf{z}^+)$ and $-\mathrm{Sum}(\mathbf{z}^-)$ are both at least equal to $k'\lceil q/2\rceil$ or both at most equal to this value. Define

$$\nu=\begin{cases}+, & \text{if } \mathrm{Sum}(\mathbf{z}^+)\geqslant-\mathrm{Sum}(\mathbf{z}^-)\geqslant k'\lceil q/2\rceil\\ & \text{or } \mathrm{Sum}(\mathbf{z}^+)\leqslant-\mathrm{Sum}(\mathbf{z}^-)\leqslant k'\lceil q/2\rceil,\\ -, & \text{otherwise.}\end{cases} \qquad (29)$$

In the second (and last) step of the modification process, we change either the positive or the negative symbols in $\mathbf{z}$, in a manner similar to that used in Subsection IV-C. Consider $\lfloor q/2\rfloor k'$ balancing sequences

$$\mathbf{b}_i=(j+2)^g j^{k'-g},$$

$i=0,1,\ldots,\lfloor q/2\rfloor k'-1$, where $j=2\lfloor i/k'\rfloor$ and $g=i-k'\lfloor i/k'\rfloor$. Throughout the rest of this subsection, let $\oplus$ denote the addition over the integer numbers, with a reduction modulo

$2\lfloor q/2 \rfloor$ such that the final outcome is in $\mathcal{A}_q^+ = \{j \in \mathcal{A}_q : j > 0\}$ if $\nu = +$ and in $\mathcal{A}_q^- = \{j \in \mathcal{A}_q : j < 0\}$ if $\nu = -$. We replace $\mathbf{z}^\nu$ by $\mathbf{z}^\nu \oplus \mathbf{b}_w$, where w is chosen such that

$$\mathrm{Sum}(\mathbf{z}^\nu \oplus \mathbf{b}_w) = -\mathrm{Sum}(\mathbf{z}^{\bar{\nu}}), \tag{30}$$

where $\bar{\nu}$ denotes the inverse of ν. In conclusion, the resulting sequence $\mathbf{x}$ satisfies (27).

In summary, we have the following algorithm in case $q \geqslant 4$.

CPB Encoding Procedure

1) Apply the encoding procedure from Subsection IV-B to change the information sequence $\mathbf{u}$ into a PB sequence $\mathbf{y}$, using appropriate offset a (if q is odd) and PB index z.
2) Compute ξ by (28).
3) If $\xi = 1$, then replace all symbols y_i in $\mathbf{y}^+$ by $2\lceil q/2 \rceil - y_i$; else leave them untouched. Call the resulting sequence $\mathbf{z}$.
4) Compute ν by (29).
5) Determine an index w such that (30) is satisfied.
6) Replace in $\mathbf{z}$ the subsequence $\mathbf{z}^\nu$ by $\mathbf{z}^\nu \oplus \mathbf{b}_w$, to obtain the CPB sequence $\mathbf{x}$, .
7) Map (z, ξ, ν, w) (if q even) or (a, z, ξ, ν, w) (if q odd) to a unique CPB prefix $\mathbf{p}$.

Then transmit or store the balanced codeword $\mathbf{c} = (\mathbf{p}, \mathbf{x})$.

CPB Decoding Procedure

1) Retrieve a (if q is odd), z, ξ, ν, and w from the prefix $\mathbf{p}$.
2) Replace $\mathbf{x}^\nu$ by $\mathbf{x}^\nu \oplus (-\mathbf{b}_w)$ in $\mathbf{x}$ to obtain $\mathbf{z}$.
3) If $\xi = 1$, then replace all symbols z_i in $\mathbf{z}^+$ by $2\lceil q/2 \rceil - z_i$; else leave them untouched. Call the resulting sequence $\mathbf{y}$.
4) Apply the decoding procedure from Subsection IV-B to retrieve $\mathbf{u}$ from $\mathbf{y}$, using a (if q is odd) and z.

Proof. It is easy to see that the operations in the encoding procedure are properly reversed in the decoding procedure. Further, the validity of the PB part was already demonstrated in Subsection IV-B. Hence, the only thing left to prove is that there always exists a suitable index w. To this end, define $\mathbf{b}_{\lfloor q/2 \rfloor k'} = \mathbf{b}_0$ and consider the series

$$\mathrm{Sum}(\mathbf{z}^\nu \oplus \mathbf{b}_0), \mathrm{Sum}(\mathbf{z}^\nu \oplus \mathbf{b}_1), \ldots, \mathrm{Sum}(\mathbf{z}^\nu \oplus \mathbf{b}_{\lfloor q/2 \rfloor k'}).$$

We make the following observations.

1) The series starts and ends with the same even value.
2) For all $i \in \{0, 1, \ldots, \lfloor q/2 \rfloor k' - 1\}$, it holds that
$$\mathrm{Sum}(\mathbf{z}^\nu \oplus \mathbf{b}_{i+1}) = \mathrm{Sum}(\mathbf{z}^\nu \oplus \mathbf{b}_i) + c,$$
where c is either $+2$ or $-2\lfloor q/2 \rfloor + 2$.
3) It holds that
$$\begin{aligned} \left| \sum_{j=0}^{\lfloor q/2 \rfloor - 1} \mathrm{Sum}(\mathbf{z}^\nu \oplus \mathbf{b}_{jk'}) \right| &= k' \sum_{j=0}^{\lfloor q/2 \rfloor - 1} (q - 1 - 2j) \\ &= \lfloor q/2 \rfloor k' \lceil q/2 \rceil. \end{aligned}$$
Hence, the average value of all $\mathrm{Sum}(\mathbf{z}^\nu \oplus \mathbf{b}_{jk'})$, with $j = 0, 1, \ldots, \lfloor q/2 \rfloor - 1$, is $k' \lceil q/2 \rceil$.

By combining these three observations and (29), we can conclude that there exists at least one w in $\{0, 1, \ldots, \lfloor q/2 \rfloor k' - 1\}$ such that (30) is satisfied. □

Since there are q possible values for a, k for z, 2 for ξ, 2 for ν, and $\lfloor q/2 \rfloor k' \leqslant \lfloor q/2 \rfloor \lfloor k/2 \rfloor$ for w, it is sufficient to choose the prefix length such that

$$P = 4k \lfloor q/2 \rfloor \lfloor k/2 \rfloor = qk^2$$

CPB sequences can be accommodated if q is even, and

$$P = 4qk \lfloor q/2 \rfloor \lfloor k/2 \rfloor = 2q(q-1)k \lfloor k/2 \rfloor$$

if q is odd. Hence, the unbalanced redundancy is

$$p' = \log_q P = 1 + 2 \log_q k$$

if q is even, and very close to that number if q is odd.

Example 4. We use the same information sequence as in Examples 2 and 3, i.e.,

$$\mathbf{u} = (+4, +4, -2, 0, 0, 0, 0) \in (\mathcal{A}_5)^7.$$

Encoding into a CPB sequence goes as follows.

1) From Example 2, the PB sequence
$$\mathbf{y} = (+4, +4, 0, -2, -2, -2, +2)$$
is obtained.
2) Find $\xi = 1$, since
$$-\mathrm{Sum}(\mathbf{y}^-) = 6 < 9 < 10 = \mathrm{Sum}(\mathbf{y}^+).$$
3) Mirror the positive values in $\mathbf{y}$ with respect to $+3$ to obtain
$$\mathbf{z} = (+2, +2, 0, -2, -2, -2, +4).$$
4) Find $\nu = -$, since
$$-\mathrm{Sum}(\mathbf{z}^-) = 6 < 8 = \mathrm{Sum}(\mathbf{z}^+) \leqslant 9.$$
5) Determine $w = 1$ as a suitable balancing index.
6) Add (modulo 4, with the resulting symbols in the set $\{-4, -2\}$) the sequence $\mathbf{b}_1 = (2, 0, 0)$ to $\mathbf{z}^-$, i.e., compute
$$\begin{aligned} \mathbf{x} &= (+2, +2, 0, -2, -2, -2, +4) \\ &\quad \oplus (0, 0, 0, 2, 0, 0, 0) \\ &= (+2, +2, 0, -4, -2, -2, +4) \end{aligned}$$
7) Uniquely map $(a, z, \xi, \nu, w) = (-2, 6, 1, -, 1)$ to one of the CPB sequences of length 6, e.g.,
$$\mathbf{p} = (+4, +2, -2, -4, +4, -4).$$

Then the CPB transmitted/stored sequence is

$$\mathbf{c} = (+4, +2, -2, -4, +4, -4, +2, +2, 0, -4, -2, -2, +4).$$

E. Symbol-Balanced Code Construction

At first sight, the Knuth approach may seem to be less suitable for generating q-ary SB sequences than for CB and PB sequences. Still, Mascella and Tallini presented Knuth-like SB construction methods which are based on maps exchanging alphabet symbols [10], [11]. By applying $q-1$ such maps, each guaranteeing that a particular symbol appears the desired number of times, symbol balancing is achieved. Here, we present another Knuth-like SB method which is similar to this Mascella-Tallini approach in the sense that it also operates in $q-1$ rounds, but is different in the sense that it adds in each round an appropriate balancing sequence to the data sequence, rather than performing specific symbol exchanges. Hence, our method is more in the spirit of the constructions presented in the previous subsections.

In order to encode a data sequence $\mathbf{u}$ of length $k = qm$ into an SB sequence $\mathbf{x}$, we propose the following Knuth-like algorithm. It consists of $q-1$ rounds, numbered 1, 2, . . ., $q-1$, in which we will perform simple reversible manipulations on the data sequence, such that the end result is SB. In round v, we force there to be exactly $m = k/q$ symbols $-q+1+2v$ in the sequence, a condition that will not change anymore in the next rounds. For $v = 1, 2, \ldots, q-1$, let

$$\mathcal{A}_q^v = \{-q-1+2v, -q+1+2v, \ldots, q-1\},$$

i.e., $\mathcal{A}_q^v$ is the sub-alphabet consisting of the $q+1-v$ largest elements of the alphabet $\mathcal{A}_q$,

$$M_v(\mathbf{y}) = \max\{j \in \mathcal{A}_q^v : S_j(\mathbf{y}) \geqslant S_i(\mathbf{y}) \ \forall i \in \mathcal{A}_q^v\}, \quad (31)$$

and

$$m_v(\mathbf{y}) = \min\{j \in \mathcal{A}_q^v : S_j(\mathbf{y}) \leqslant S_i(\mathbf{y}) \ \forall i \in \mathcal{A}_q^v\}, \quad (32)$$

where $\mathbf{y}$ is a sequence over the alphabet $\mathcal{A}_q$. Note that, for all v, $M_v(\mathbf{y})$ is a symbol from $\mathcal{A}_q^v$ appearing most frequently in $\mathbf{y}$, while $m_v(\mathbf{y})$ is a symbols from $\mathcal{A}_q^v$ appearing least frequently in $\mathbf{y}$.

The algorithm is described as follows.

SB Encoding Procedure

1) Set $\mathbf{u}_0 = \mathbf{u}$ and $v = 1$.
2) Set $m_v = m_v(\mathbf{u}_{v-1})$, $M_v = M_v(\mathbf{u}_{v-1})$, and create $\mathbf{u}_v$ from $\mathbf{u}_{v-1} = (h_1, h_2, \ldots, h_k)$ by leaving all $h_i \notin \mathcal{A}_q^v$ unchanged and adding the value
$$\begin{cases} -q-1+2v-m_v & \text{if } i \leqslant i_v, \\ -q-1+2v-M_v & \text{if } i > i_v, \end{cases} \quad (33)$$
to the $h_i \in \mathcal{A}_q^v$. The addition is done modulo $2q+2-2v$ such that the resulting symbol is in $\mathcal{A}_q^v$. The value $i_v \in \{0, 1, \ldots, k\}$ is chosen such that
$$S_{-q-1+2v}(\mathbf{u}_v) = k/q = m. \quad (34)$$
3) If $v < q-1$, then increase v by one and go back to the previous step.
4) Set $\mathbf{x} = \mathbf{u}_{q-1}$, which is SB, and map
$$(i_1, \ldots, i_{q-1}, m_1, \ldots, m_{q-1}, M_1, \ldots, M_{q-1})$$
to a unique SB prefix $\mathbf{p}$.

Then transmit or store the SB codeword $\mathbf{c} = (\mathbf{p}, \mathbf{x})$.

SB Decoding Procedure

1) Retrieve
$$(i_1, \ldots, i_{q-1}, m_1, \ldots, m_{q-1}, M_1, \ldots, M_{q-1})$$
from $\mathbf{p}$ and set $\mathbf{x}_q = \mathbf{x}$ and $v = q-1$.
2) Create $\mathbf{x}_v$ from $\mathbf{x}_{v+1} = (h_1, h_2, \ldots, h_k)$ by leaving all $h_i \notin \mathcal{A}_q^v$ unchanged, and subtracting the value as given in (33) from the $h_i \in \mathcal{A}_q^v$. The subtraction is done modulo $2q+2-2v$ such that the resulting symbol is in $\mathcal{A}_q^v$.
3) If $v > 1$, then decrease v by one and go back to the previous step.
4) Set $\mathbf{u} = \mathbf{x}_1$.

Proof. By construction we have

$$S_{-q-1+2v}(\mathbf{u}_w) = S_{-q-1+2v}(\mathbf{u}_v)$$

for all $1 \leqslant v < w \leqslant q-1$, and thus it follows from (34) that all symbols from $\mathcal{A}_q$ appear equally often in $\mathbf{x} = \mathbf{u}_{q-1}$, and thus $\mathbf{x}$ is SB. Further, it is easy to see that the operations in the encoding procedure are properly reversed in the decoding procedure. Hence, the only thing left to show is that for all $v = 1, 2, \ldots, q-1$ there always exists at least one i_v such that (34) is satisfied. From (31) and (32), it follows that $S_{m_1}(\mathbf{u}_0) \leqslant m \leqslant S_{M_1}(\mathbf{u}_0)$, and thus

$$S_{-q+1}(\mathbf{u}_1) = S_{M_1}(\mathbf{u}_0) \geqslant m \text{ if } i_1 = 0,$$

while

$$S_{-q+1}(\mathbf{u}_1) = S_{m_1}(\mathbf{u}_0) \leqslant m \text{ if } i_1 = k.$$

Since increasing or decreasing i_1 by 1 increases $S_{-q+1}(\mathbf{u}_1)$ by -1, 0, or $+1$, we can conclude that $S_{-q+1}(\mathbf{u}_1) = m$ for at least one $i_1 \in \{0, 1, \ldots, k\}$. Similarly, for $v > 1$, we have

$$S_{m_v}(\mathbf{u}_{v-1}) \leqslant \frac{k-(v-1)m}{q+1-v} = m \leqslant S_{M_v}(\mathbf{u}_{v-1}),$$

and thus

$$S_{-q-1+2v}(\mathbf{u}_v) = S_{M_v}(\mathbf{u}_{v-1}) \geqslant m \text{ if } i_v = 0,$$

while

$$S_{-q-1+2v}(\mathbf{u}_v) = S_{m_v}(\mathbf{u}_{v-1}) \leqslant m \text{ if } i_v = k,$$

and so $S_{-q-1+2v}(\mathbf{u}_v) = m$ for at least one value $i_v \in \{0, 1, \ldots, k\}$. □

Note that there are at most $(k+1)^{q-1}$ possible realizations of $(i_1, \ldots, i_{q-1})$, $q!$ possible realizations of $(m_1, \ldots, m_{q-1})$, and $q!$ possible realizations of $(M_1, \ldots, M_{q-1})$. Hence, an unbalanced redundancy of

$$p' = (q-1)\log_q(k+1) + 2\log_q(q!)$$

suffices. We conclude that, as for the Mascella-Tallini constructions [10], [11], the redundancy of this Knuth-like SB method exceeds the minimum redundancy by a factor of two for long codes.

Example 5. Let $q = 3$ and $n = 6$, and thus the symbol frequency should be $m = 6/3 = 2$. The data sequence is given to be

$$\mathbf{u} = \mathbf{u}_0 = (0, -2, -2, -2, 0, -2).$$

Hence, $S_{-2}(\mathbf{u}_0) = 4$, $S_0(\mathbf{u}_0) = 2$, $S_{+2}(\mathbf{u}_0) = 0$, and thus $M_1 = -2$ (the most frequent symbol) and $m_1 = +2$ (the least frequent symbol). According to (33), in the first round ($v = 1$), the number of -2 symbols is forced to be 2 by modulo-6 adding -4 to the first i_1 symbols of $\mathbf{u}_0$ and 0 to the last $6 - i_1$ symbols. Choosing $i_1 = 3$ gives

$$\mathbf{u}_1 = (+2, 0, 0, -2, 0, -2).$$

Note that $S_{-2}(\mathbf{u}_1) = 2$, $S_0(\mathbf{u}_1) = 3$, $S_{+2}(\mathbf{u}_1) = 1$, and thus $M_2 = 0$ and $m_2 = +2$. In the next round ($v = 2$), the number of zeroes is forced to be 2 by modulo-4 adding -2 to the first i_2 symbols of $\mathbf{u}_1$ and 0 to the last $6 - i_2$ symbols, except when the symbol is equal to -2, in which case we leave it unchanged. Choosing $i_2 = 3$ gives

$$\mathbf{u}_2 = (0, +2, +2, -2, 0, -2).$$

Note that $S_0(\mathbf{u}_2) = S_1(\mathbf{u}_2) = S_2(\mathbf{u}_2) = 2$, and thus $\mathbf{x} = \mathbf{u}_2$ is SB.

F. Discussion

In the previous subsections, we have presented generalizations of Knuth's binary/bipolar balancing algorithm to larger alphabets, for the various balancing perspectives under consideration in this paper. Examples have been provided to illustrate the (encoding) procedures. It should be mentioned that these examples are misleading in the sense that the redundancy appears to be relatively large, which is due to the fact that extremely short data blocks were used in the examples. For instance, in Example 2, four redundant symbols are used for eight data symbols. However, for long codes, the redundancy is only logarithmic in the length of the data block. For all the constructions presented in this section, the redundancy is roughly twice the corresponding minimum redundancy derived in Section III.

For the binary case, modifications of Knuth's method have been presented to close the factor of two gap between the redundancy of the original Knuth algorithm and the minimum redundancy, while maintaining sufficient simplicity to enable feasible implementations. In [8], this is done by a more efficient (variable-length) encoding of the prefix. In [21], minimum redundancy is achieved by exploiting the fact that many data sequences have more than one possible balancing index, thus allowing to encode auxiliary data through the choice of the index. It is an interesting research challenge to investigate whether such techniques are also applicable in non-binary cases.

V. Conclusions

In this paper we have considered balancing of q-ary sequences from various perspectives. In particular, we have reviewed the symbol balancing and charge balancing concepts, and introduced the polarity balancing concept, also in combination with charge balancing. For each of these perspectives, we have derived (approximate) expressions for the number of such sequences of a fixed length and for the minimum redundancy. The major conclusions of this analysis have been summarized in Table II, which shows the minimum redundancy normalized to the logarithm of the block length n in the limit as $n \to \infty$. Furthermore, we have presented for each of the balancing perspectives a q-ary coding scheme in the spirit of the binary Knuth algorithm. These schemes allow for simple encoding and decoding, at the price of a redundancy which is twice the minimum required redundancy.

References

[1] S. Al-Bassam and B. Bose, "On balanced codes," *IEEE Trans. Inf. Theory*, vol. 36, no. 2, pp. 406–408, Oct. 1993.

[2] S. Al-Bassam and B. Bose, "Design of efficient error-correcting balanced codes," *IEEE Trans. Comp.*, vol. 42, no. 10, pp. 1261–1266, Oct. 1993.

[3] S. Al-Bassam and B. Bose, "Design of efficient balanced codes," *IEEE Trans. Comp.*, vol. 43, no. 3, pp. 362–365, Mar. 1994.

[4] N. Alon, E. E. Bergmann, D. Coppersmith and A. M. Odlyzko, "Balancing sets of vectors," *IEEE Trans. Inf. Theory*, vol. 34, no. 1, pp. 129–130, Jan. 1988.

[5] A. Baliga and S. Boztaş, "Balancing sets of non-binary vectors", *Proc. IEEE Int. Symp. Inform. Theory*, Lausanne, Switzerland, p. 300, June 30–July 5, 2002.

[6] R. M. Capocelli, L. Gargano and U. Vaccaro, "Efficient q-ary immutable codes," *Discrete Applied Mathematics*, vol. 33, pp. 25–41, 1991.

[7] K. A. S. Immink, *Codes for Mass Data Storage Systems,* Second Edition, Shannon Foundation Publishers, Eindhoven, The Netherlands, 2004.

[8] K. A. S. Immink and J. H. Weber, "Very efficient balanced codes", *IEEE J. Sel. Areas Commun.*, vol. 28, no. 2, pp. 188–192, Feb. 2010.

[9] D. E. Knuth, "Efficient balanced codes," *IEEE Trans. Inf. Theory*, vol. 32, no. 1, pp. 51–53, Jan. 1986.

[10] R. Mascella and L. G. Tallini, "On symbol permutation invariant balanced codes," *Proc. IEEE Int. Symp. Inform. Theory*, Adelaide, Australia, pp. 2100–2104, Sept. 4–9, 2005.

[11] R. Mascella and L. G. Tallini, "Efficient m-ary balanced codes which are invariant under symbol permutation," *IEEE Trans. Comp.*, vol. 55, no. 8, pp. 929–946, Aug. 2006.

[12] R. Mascella, L. G. Tallini, S. Al-Bassam and B. Bose, "On efficient balanced codes over the mth roots of unity," *IEEE Trans. Inf. Theory*, vol. 52, no. 5, pp. 2214–2217, May 2006.

[13] A. Mazumdar, R. M. Roth, and P. O. Vontobel, "On linear balancing sets", *Proc. IEEE Int. Symp. Inform. Theory*, Seoul, South Korea, pp. 2699–2703, June 28–July 3, 2009.

[14] D. Pelusi, L. G. Tallini, and B. Bose, "On m-ary balanced codes with parallel decoding," *Proc. IEEE Int. Symp. Inform. Theory*, Austin, Texas, USA, pp. 1305–1309, June 13–18, 2010.

[15] Z. Star, "An asymptotic formula in the theory of compositions," *Aequationes Mathematicae*, vol. 13, pp. 279–284, 1975.

[16] T. G. Swart and J. H. Weber, "Efficient balancing of q-ary sequences with parallel decoding," *Proc. IEEE Int. Symp. Inform. Theory*, Seoul, South Korea, pp. 1564–1568, June 28–July 3, 2009.

[17] L. G. Tallini and B. Bose, "Balanced codes with parallel encoding and decoding," *IEEE Trans. Comp.*, vol. 48, no. 8, pp. 794–814, Aug. 1999.

[18] L. G. Tallini, R. M. Capocelli and B. Bose, "Design of some new efficient balanced codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 3, pp. 790–802, May 1996.

[19] L. G. Tallini and U. Vaccaro, "Efficient m-ary balanced codes", *Discrete Applied Mathematics*, vol. 92, pp. 17–56, 1999.

[20] H. van Tilborg and M. Blaum, "On error-correcting balanced codes", *IEEE Trans. Inf. Theory*, vol. 35, no. 5, pp. 1091–1095, Sept. 1989.

[21] J. H. Weber and K. A. S. Immink, "Knuth's balanced codes revisited", *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1673–1679, Apr. 2010.

[22] J. H. Weber, K. A. S. Immink, and H.C. Ferreira, "Error-correcting balanced Knuth codes", *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 82–89, Jan. 2012.